REMARKS

The Examiner has rejected Claims 1-4, 9-12 and 15 under 35 U.S.C. 101 for claiming the same invention as that of Claims 1-4, 5-8 and 9, respectively, in US Patent No. 6,593,923. Further, the Examiner has rejected Claims 30, 31, 33, 34, 36, 37, 39, 40, and 44 under 35 U.S.C. 101 for claiming the same invention as that of Claims 7-10 and 17-20, respectively, of US Patent No. 6,690,372.

Applicant respectfully disagrees with such rejection, especially when applicant's claims are read in view of the amendments made to the independent claims in the previous amendments submitted by applicant. Specifically, in such previous amendment, applicant broadened independent Claims 1, 10 and 15 such that they are not identical in claim scope with respect to the foregoing patent relied on by the Examiner. To clearly demonstrate the differences between such claims, applicant has attached a claim chart (see Claim Chart A) comparing applicant's claims, as currently pending, to the relevant claims of US Patent No. 6,593,923.

Furthermore, independent Claim 29 et al. has been clarified via a previously amendment such that it also is not identical in scope with respect to the relevant claims of US Patent No. 6,690,372. Specifically, the claims of US Patent No. 6,690,372 do not require a "single shader unit" as now claimed, and the instant claims do not require the following limitations required by one or more claims in US Patent No. 6,690,372:

> "the first and second shading calculations together include a plurality of decoupled variables" (see Claims 1, 11, and 21), "wherein the first shading calculation includes [((1-s))*(Color_diff+Color_spec)]for generating an output A, and the second shading calculation includes [Color_amb+A], where s is a shadow variable, Color_diff is a diffuse color variable Color_spec is a specular color variable, and Color_amb is an ambient color variable" (see Claims 7 and 17), "wherein the first shading calculation includes [(1-

s)*(Color_diff+Color_spec)]for generating an output A, and the second shading calculation includes [A*Texture_det+(1-s)* Color_spec], where s is a shadow variable Color_duff is a diffuse color variable, Color_spec is a specular color variable, Color_amb is an ambient color variable, and Texture_det is a detail texture variable" (see Claim 8), "wherein the first shading calculation includes [((1-s)* Color_diff)+Color_amb]for generating an output A, and the second shading calculation includes [A*Texture_det+((1-s))* Color_spec], where s is a shadow variable Color_diff is a diffuse color variable, Color spec is a specular color variable Color_amb is an ambient color variable, and Texture_det is a texture detail variable" (see Claim 18), "wherein [the] first and second shading calculations together include a diffuse color variable, a specular color variable, and an ambient color variable; wherein the variables are decoupled" (see Claims 9 and 19), and "(a) a shading module for performing the first shading calculation in order to generate the output; (b) a texture look-up module coupled to the shading module for retrieving texture information using texture coordinates associated with the output; (c) a feedback loop coupled between an input and an output of the shading module for performing the second shading calculation using the texture information from the texture look-up module in order to generate further output; and (d) a combiner module coupled to the output of the shading module for combining the output generated by the shading module" (see Claims 10 and 20).

In view of the clear differences between applicant's independent claims and those of the referenced patents, applicant respectfully asserts that the associated dependent claims are therefore also not identical in scope when read in view of their dependence on such independent claims.

The Examiner has rejected Claims 1-4, 9-12, 15, 29, 32, 35, 38 and 41-43 under 35 U.S.C. 102(a) as being anticipated by Woo et al. (Open GL Programming Guide, 3rd edition, Mason Woo, Silicon Graphics 1999).

With respect to independent Claims 1, 10 and 15, the Examiner has relied on Woo, pages 251-253, to make a prior art showing of applicant's claimed "conditionally clamping the depth value based on the value of the slope" (see this or similar, but not identical, language in each of the foregoing claims). Applicant respectfully disagrees with such rejection. Specifically, Woo expressly teaches that the "depth values are in window coordinates, clamped to the range [0,1]" (see page 251, paragraph 4 line 6). Woo does not elaborate on how this is accomplished. Applicant, on the other hand, claims that the depth value is "conditionally" clamped "based on the value of the slope," which clearly departs from the static range taught by Woo.

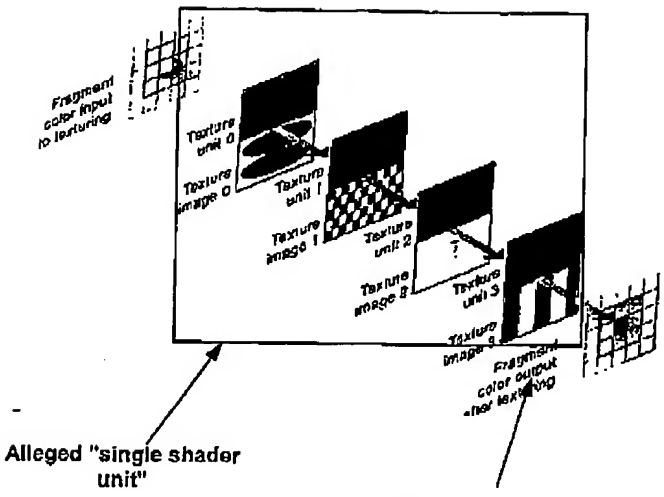
With respect to independent Claims 29, 35, and 41, the Examiner has submitted the following arguments regarding the "single shader unit" language set forth in the exemplary claim below.

"performing a first shading calculation in order to generate output utilizing a single shader unit of a graphics pipeline;

saving the output; and

performing a second shading calculation using the output in order to generate further output utilizing the single shader unit of the graphics pipeline" (emphasis added - see this or similar, but not identical, language in independent Claim 35 et al.).

In particular, the Examiner argues that "the shader unit composites the various texture images, thus there is only one shader with multiple texture units." It thus appears that the Examiner is arguing that the multiple texture units shown below in Woo constitute a single shader unit. To this end, applicant has encircled such multiple texture units to show what the Examiner appears to deem a single shader unit.



However, even if this assumption is made, the above disclosure in Woo fails to meet applicant's claims. Specifically, the above figure clearly shows the output of the alleged single shader unit. Unfortunately, however, such output is not subject to "a second shading calculation using the output in order to generate further output utilizing the single shader unit of the graphics pipeline" (emphasis added), as claimed.

Applicant respectfully asserts that this clear deficiency in Woo is rooted in the fact that Woo's series of texture units simply fails to meet applicant's single shader unit, which provides for a more cost-effective design that is still capable of performing multiple shading operations.

The Examiner is reminded that a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described in a single prior art reference. *Verdegaal Bros. v. Union Oil Co. Of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). Moreover, the identical invention must be shown in as complete detail as contained in the claim. *Richardson v. Suzuki*

*Motor Co.*868 F.2d 1226, 1236, 9USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim.

This criterion has simply not been met by the Woo reference. A notice of allowance or a specific prior art showing of each of the foregoing claimed features, in combination with the remaining claimed features, is respectfully requested.

Applicant further notes that the prior art is also deficient with respect to the dependent claims. For example, with respect to dependent Claim 3 et al., the Examiner has relied on Woo's disclosure of using a larger offset if "the depth slope [is] greater than zero" such that a "small, nonzero for *factor*, such as 0.75 or 1.0" is used (page 252). Applicant respectfully asserts that using a larger offset when the slope is greater than zero does not meet applicant's specific claim language. Woo clearly fails to teach "wherein the depth value is clamped if the value of the slope is greater than a predetermined amount" (emphasis added), as claimed by applicant, since Woo merely teaches using a larger offset if the depth slope is greater than zero.

With respect to dependent Claim 4 et al., the Examiner has relied on Woo's disclosed "depth value of each fragment [that] is added to a calculated offset value" (page 251) to make a prior art showing of applicant's claimed "wherein the clamping includes the steps of: identifying vertex depth values of vertices of a primitive; comparing at least one of the vertex depth values with the depth value generated by the offset operation; and clamping the depth value generated by the offset operation based on the comparison." Applicant respectfully asserts that simply nowhere in Woo is there any teaching of "comparing at least one of the vertex depth values with the depth value generated by the offset operation" (emphasis added), as claimed by applicant, since Woo merely teaches calculating an offset value and adding the depth value of each fragment to the calculated offset value.

A notice of allowance or a specific prior art showing of all of applicant's claim limitations, in combination with the remaining claim elements, is respectfully requested.

To this end, all of the independent claims are deemed allowable. Moreover, the remaining dependent claims are further deemed allowable, in view of their dependence on such independent claims.

In the event a telephone conversation would expedite the prosecution of this application, the Examiner may reach the undersigned at (408) 505-5100. If any fees are due in connection with the filing of this paper, then the Commissioner is authorized to charge such fees to Deposit Account No. 50-1351 (Order No. NVIDP030A).

Respectfully submitted,

Kevin J. Zilka
Registration No. 41,429

P.O. Box 721120
San Jose, CA 95172
Telephone: (408) 505-5100

CLAIM CHART A

Claims of U.S. App. No.: 10/676,788 *emphasis has been added to emphasize the distinctions.	U.S. Patent No.: 6,593,923
1. A method for shadow mapping: comprising: performing an offset operation to generate a depth value; identifying a value of a slope; and conditionally clamping the depth value based on the value of the slope.	1. A method for shadow mapping while rendering a primitive in a graphics pipeline, comprising: (a) performing an offset operation to generate a depth value while rendering a primitive; (b) identifying a value of a slope associated with a primitive; and (c) conditionally clamping the depth value based on the value of the slope.
10. A computer program computer program embodied on a computer readable medium for shadow mapping, comprising: a code segment for performing an offset operation to generate a depth value; a code segment for identifying a value of a slope; and a code segment for conditionally clamping the depth value based on the value of the slope.	6. A computer program embodied on a computer readable medium for shadow mapping while rendering a primitive in a graphics pipeline, comprising: (a) a code segment for performing an offset operation to generate a depth value while rendering a primitive; (b) a code segment for identifying a value of a slope associated with a slope of the primitive; and (c) a code segment for conditionally clamping the depth value based on the value of the slope.
15. A system for shadow mapping, comprising: logic for performing an offset operation to generate a depth value; logic for calculating and identifying a value of a slope; and logic for conditionally clamping the depth value based on the value of the slope.	9. A system for shadow mapping while rendering a primitive in a graphics pipeline, comprising: (a) logic for performing an offset operation to generate a depth value while rendering a primitive; (b) logic for calculating and identifying a value of a slope associated with the primitive; and (c) logic for conditionally clamping the depth value based on the value of the slope.